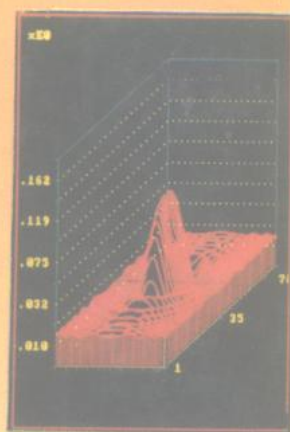
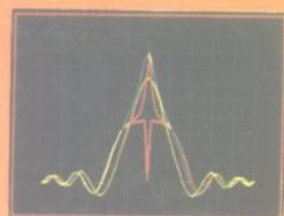


《现代电子技术》增刊

计算机 三维图形常用算法 与

王 冰
潘建寿 编著



C 语言 程序

陕西电子杂志社

7/31
W
385723

计算机三维图形常用算法 与 C 语言程序

王 冰 潘建寿 编 著

TP391.4
713

陕西电子杂志社

内容提要

本书详细讨论了绘制三维图形的原理、算法及实施步骤。内容包括:三维建模,模型处理、隐面清除、涂色、光照及阴影等。在讨论原理的基础上,本书重点放在算法及程序编写上。介绍了大量的三维绘图编程的方法和技巧。书中给出的 TurboC 和 QuickC 的源程序。全部编译运行通过,并且能适用于各种微机所配备的不同的图形适配器和显示器。

作者

一九九四年十一月

目 录

第一章 三维图象基本概念

§ 1.1	世界坐标	(1)
§ 1.2	建立三维模型	(2)
§ 1.3	模型组件	(2)
§ 1.4	边界描述模型	(3)
§ 1.5	帧相模型	(3)
§ 1.6	基础构件	(4)
§ 1.7	立体几何建造模型	(4)
§ 1.8	显示选择	(7)

第二章 三维模型处理

§ 2.1	三维算法	(8)
§ 2.2	旋转公式	(8)
§ 2.3	平移公式	(9)
§ 2.4	投影公式	(11)
§ 2.5	角形变	(11)
§ 2.6	度和弧度	(14)
§ 2.7	旋转和平移次序	(14)
§ 2.8	放大和缩小	(16)
§ 2.9	模型建造	(16)

第三章 立方体

§ 3.1	框线立方体	(18)
§ 3.2	全局变量和局部变量	(19)
§ 3.3	主函数	(19)
3.3.1	子程序 graphics—setup()	(19)
3.3.2	主程序	(20)
§ 3.4	数据初始化	(21)
§ 3.5	隐面消除	(22)
§ 3.6	径向预分类法	(22)
§ 3.7	径向分类法	(22)
§ 3.8	平面方程方法	(22)

§ 3.9	物空间和象空间	(23)
§ 3.10	立方体;使用平面方程方法	(23)
§ 3.11	平面方程公式	(24)
§ 3.12	优化数据库	(24)
§ 3.13	使用 Key matte	(25)
§ 3.14	一种通用算法	(26)
§ 3.15	编译注意	(27)
§ 3.16	演示程序 Quick C - 01	(27)
§ 3.17	演示程序 Turbo C - 01	(33)
§ 3.18	演示程序 Quick C - 02	(40)
§ 3.19	演示程序 Turbo C - 02	(49)

第四章 球体和圆柱体

§ 4.1	球体	(59)
4.1.1	程序要点	(61)
4.1.2	区域填充起始点	(62)
4.1.3	显示器调节	(63)
§ 4.2	圆柱体	(63)
4.2.1	圆柱体产生	(63)
4.2.2	程序要点	(63)
§ 4.3	演示程序 Quicke - 03	(65)
§ 4.4	演示程序 Turboc - 03	(77)
§ 4.5	演示程序 Quicke - 04	(89)
§ 4.6	演示程序 Turboc - 04	(98)

第五章 三维空间曲线

§ 5.1	三次参数曲线	(108)
§ 5.2	三维自由格式曲线	(109)
§ 5.3	程序要点	(110)
§ 5.4	演示程序 Quick C - 05	(111)
§ 5.5	演示程序 Turbo C - 05	(118)

第六章 贴图

§ 6.1	表面贴图	(127)
§ 6.2	立方体贴图	(127)
§ 6.3	圆柱体贴图	(127)

§ 6.4	程序要点	(129)
§ 6.5	演示程序 QuickC - 06	(129)
§ 6.6	演示程序 TurboC - 06	(142)
§ 6.7	演示程序 QuickC - 07	(154)
§ 6.8	演示程序 TurboC - 07	(166)

第七章 多模型显示

§ 7.1	隐面消除方法分类	(180)
7.1.1	物空间方法	(180)
7.1.2	象空间方法	(180)
§ 7.2	隐面消除的十种方法	(180)
7.2.1	径向预分类法	(180)
7.2.2	径向分类法	(181)
7.2.3	平面方程法	(182)
7.2.4	平面分隔法	(182)
7.2.5	深度分类法	(183)
7.2.6	光线跟踪法	(183)
7.2.7	分解法	(183)
7.2.8	z - buffer 法	(184)
7.2.9	最大最小法	(184)
7.2.10	线扫描法	(185)
§ 7.3	隐面消除方法比较	(185)
§ 7.4	最适宜 C 语言的方法	(186)

第八章 照明与涂色

§ 8.1	涂色算法	(187)
8.1.1	分级涂色	(187)
8.1.2	无级涂色	(187)
§ 8.2	光学依据	(187)
8.2.1	光线入射角	(188)
8.2.2	照明和反射	(188)
8.2.3	物体的颜色	(190)
8.2.4	反射, 投影与射线跟踪法	(190)
§ 8.3	照明和涂色在个人计算机上的实现	(191)

第九章 PC 涂色模式

§ 9.1	模式构造	(193)
§ 9.2	分级涂色能力演示	(195)
§ 9.3	位填充涂色	(197)
§ 9.4	边界线消除	(198)
§ 9.5	演示程序 QuickC - 08	(198)
§ 9.6	演示程序 TurboC - 08	(208)

第十章 计算机控制涂色

§ 10.1	算法	(217)
10.1.1	通用算法	(217)
10.1.2	特殊算法	(218)
§ 10.2	子程序	(219)
10.2.1	illumination()子程序	(220)
10.2.2	shade()子程序	(223)
10.2.3	dither()子程序	(224)
§ 10.3	举例	(224)
10.3.1	立方体	(224)
10.3.2	球体	(224)
10.3.3	圆柱体	(225)
§ 10.4	演示程序 QuickC - 09	(226)
§ 10.5	演示程序 TurboC - 09	(241)
§ 10.6	演示程序 QuickC - 10	(254)
§ 10.7	演示程序 TurboC - 10	(270)
§ 10.8	演示程序 QuickC - 11	(285)
§ 10.9	演示程序 TurboC - 11	(300)

第十一章 反射和阴影

§ 11.1	镜反射	(314)
§ 11.2	几何分析	(315)
§ 11.3	阴影	(315)
§ 11.4	程序要点	(317)
§ 11.5	演示程序 Quick C - 12	(317)
§ 11.6	演示程序 Turbo C - 12	(333)

附录 A	C 语言图形数学	(349)
------	----------------	-------

第一章 三维图象基本概念

现实世界的物体都是三维的，通常用高度、宽度和深度来表示。而计算机的显示屏幕是二维的，在二维的屏幕上模拟真实物体的高度、宽度和深度所成的象，称之为三维图象。产生三维图象的过程，称之为建立模型，简称建模。给模型加上明暗、图案、阴影等，称之为涂色。

虽然建模和涂色是密切相关的，但是在作三维图象时，总是要先建立模型，然后再进行着色。本书的 1—7 章将阐述建立模型的方法和技术。第 8—11 章阐述在已有模型的基础上，给图象加上颜色，并在它的不同部分加上适当的亮度，使其有明暗感。如果再加上光源，还将产生阴影，增强象的三维效果。

§ 1.1 世界坐标

为使屏幕上显示的物体具有三维感，首先要确定描述物体各部分的位置。通常用三维直角坐标系即 xyz 坐标系。习惯上，是用 x 轴表示物体的宽度（左右）， y 轴表示物体的高度（上下）， z 轴表示物体的深度（远近），如图 1-1 所示。观察点选在坐标系的原点 $(0, 0, 0)$ ，通常，假定你就处在观察点，作为观察者。

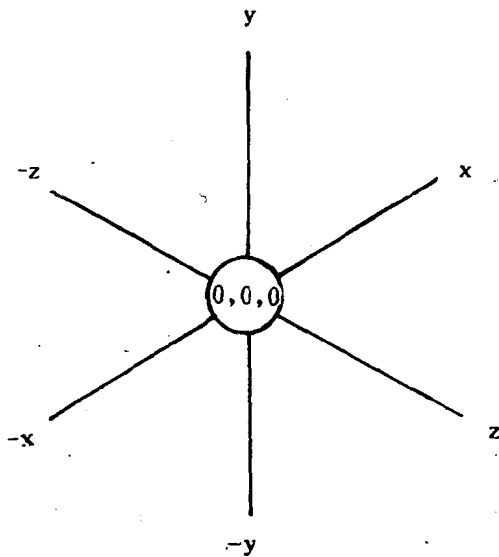


图 1—1

不论模型的形状如何，模型上的任一点都和三维物体的一点相对应，而物体的任一点，都可以用 xyz 坐标来确定，所以，每一点的 xyz 坐标都称作它的世界坐标。

建立模型的第一步就是确定物体的世界坐标。世界坐标也叫做绝对坐标，或模型空间坐标。可以说，物体的世界坐标是模型的基础，它形成模型设计的数据库。由于数据库中

的任一 xyz 坐标确定一矢量的终点。它们也被称作矢量数据。

表示真实世界的 xyz 轴体系和编写的程序没有关系。与使用哪一种计算机也没有关系。当然，物体在三维体系中的 xyz 坐标也与计算机的型号、图形适配器、屏幕分辨率等没有关系。所以，它们也被称作独立于硬件设备的坐标。

§ 1.2 建立三维模型

产生三维模型可分为三步，如图 1—2 所示。第一步，物体在三维坐标系中的一系列世界坐标确定了三维模型的基本形状。每二步，这个基本模型经过适当的平移和旋转达到新的位置，产生观察坐标。第三步，此模型被投影到平面上，投影公式将观察坐标转化为平面坐标，即 xy 平面显示坐标。

xyz 世界坐标和 xyz 观察坐标都是独立于硬件设备的坐标。它们与显示器，图形模式等都没有关系。然而， xy 平面显示坐标是和硬件设备有关系的。在按 xy 坐标在屏幕上作图时，必须考虑到所使用显示器及相应的显示模式，如是 320×200 模式，或是 640×200 模式，或是 640×480 模式。坐标是由经平移和旋转了的一系列 xyz 坐标组成，而屏幕坐标是由一系列的 xy 平面坐标构成（因为在显示屏幕上，我们只需使用 x 坐标和 y 坐标画出屏幕上的点）。

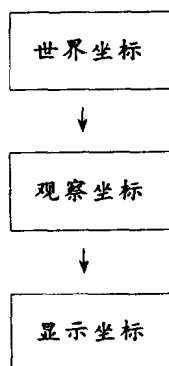


图 1—2

§ 1.3 模型组件

任一个三维模型都含有一定的组件，如图 1—3 所示。这些组件通常有顶点（线段相交形成的点）和表面（模型上的某一区域）。表面可能是平面，也可能是曲面；可能是矩形，也可能是多边形。由多边形构成的曲面也叫多边形面元。它常用来构造复杂的模型。如果一个表面只能在确定的方向观察到，而在现在的观察点观察不到，该面叫做隐面。确定一个面是否隐面的算法，叫做隐面算法。同样的，模型中不可见的线称作隐线。三维模型的任何一线也叫做三维模型的边缘。

三维模型中最重要的组件是顶点和表面。只要知道了顶点坐标，就可以画出模型，进

而确定模型中各个面的方位。在此基础上，根据光源的位置，就可以确定各个面的明暗程度，为涂色过程打下基础。利用顶点来表示整个线的方法叫做矢量方法，也叫做图形学方法。

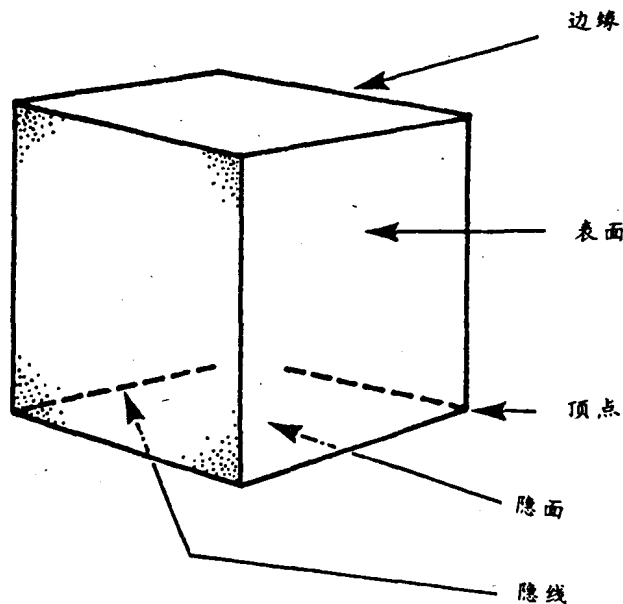


图 1—3

§ 1.4 边界描述模型

由多边形表面建造模型叫做边界描述模型。英文称作 B - pre 模型。B - pre 是英文 boundary representation 的缩写。正是由于只用三维物体的外边界来确定模型，所以叫边界描述模型或外形描述模型。边界描述模型也常称作表面模型，多边形模型，多边形网孔模型和边界模型。

§ 1.5 帧相模型

图 1 - 4 显示产生模型的三个步骤。图中只有一个实物（当然也不局限于一个物体）。物体原始的世界坐标，能产生出物体相对观察点在不同位置的不同的象，就如同摄影机对物体在不同位置可拍摄出多幅图象一样。因为每一幅图象称为一帧，所以这种模型称为帧相模型。这种模型常常用在三维内部设计程序中。比如，要显示房间里的一件家具，我们只要确定这件家具的世界坐标，就能在房间任意位置显示此家具，而且可以以任何方位来显示家具，即不论家具是正立的，还是躺倒的，是正向我们，还是背向我们，都能正确地将其显示。

§ 1.6 基础构件

在实际建模中, 常常用一系列简单的物体构件来建造三维模型。这些简单的物体构件大都是常见的几何体, 如长方体, 圆柱体, 圆锥体及球体等。在一些三维 CAD 程序或软件中, 提供有大量的基础构件, 这种方法得到很普遍地应用。

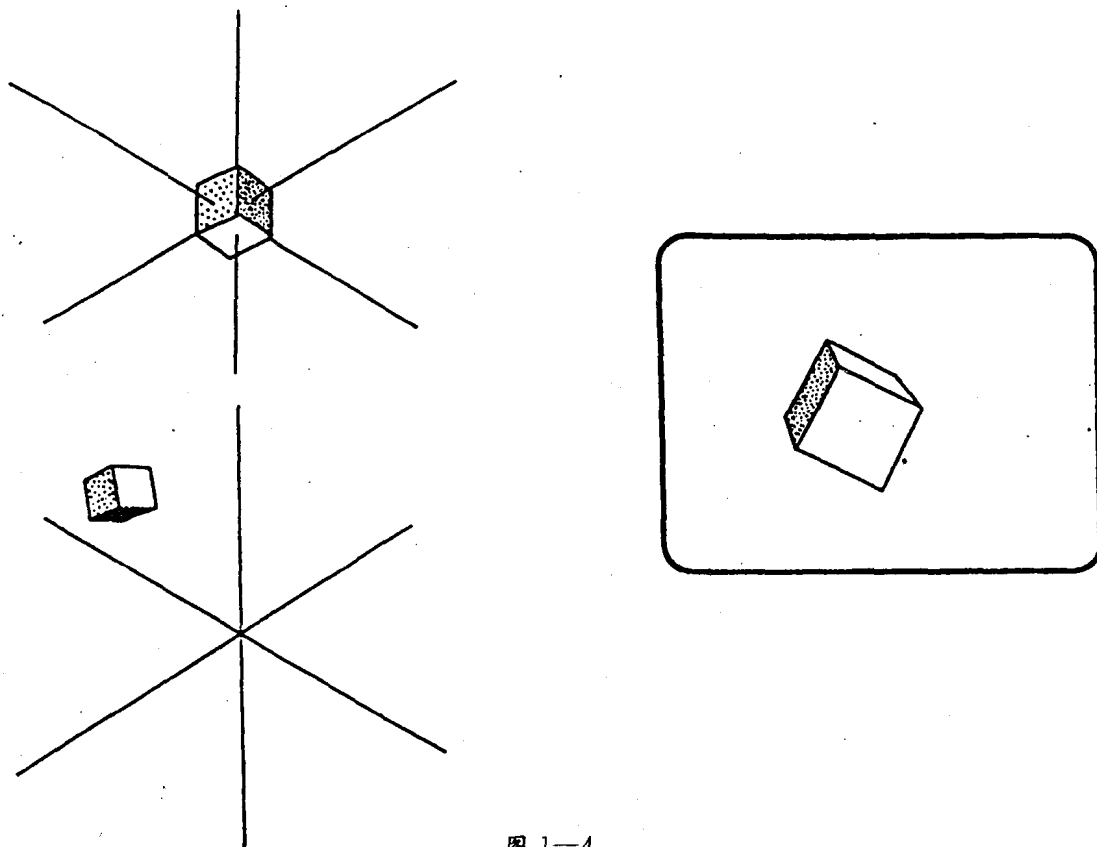


图 1—4

§ 1.7 立体几何建造模型

由基础构件建造的模型的过程称为立体几何建造模型, 英文叫 CSG 模型。CSG 是英文 constructive solid geometry 的缩写。当人们需要从模型了解其质量, 转动惯量, 质心, 体积, 表面积及其它类似工程计算时, 这种模型是特别有用的。在三维程序中, 常常需要检验以保证不会有两个物体占有同一空间的情形, 这时, 用立体几何建造模型比较好, 而此时边界描述模型是难以胜任的。立体几何建造模型在基础构件的布尔运算中也是特别有用的, 比如建造的模型要求在立方体中挖一个圆柱形的洞。布尔运算通常包括并集运算, 差集运算和交集运算。

立体几何建造方法, 也常被称作组合法, 块建造方法, 体积方法等。图 1—5 显示

边界描述和实体几何这二种方法在建模上的不同。

许多高效的三维动画程序及 CAD 软件，都将这二种建模方法结合起来使用，取各自方法的长处，以求取得最好的效果。边界描述方法适用于建造具有复杂表面物体的模型。它比较直观，而且类似于传统的画算图的方法，但在求质量、体积、质心等计算中，边界描述法效率较低，而在此情形下，正是立体几何建造法的长处。一般说来，后一种方法建造模型较好，更具有立体感。该方法的缺点是计算量大，计算耗时长。

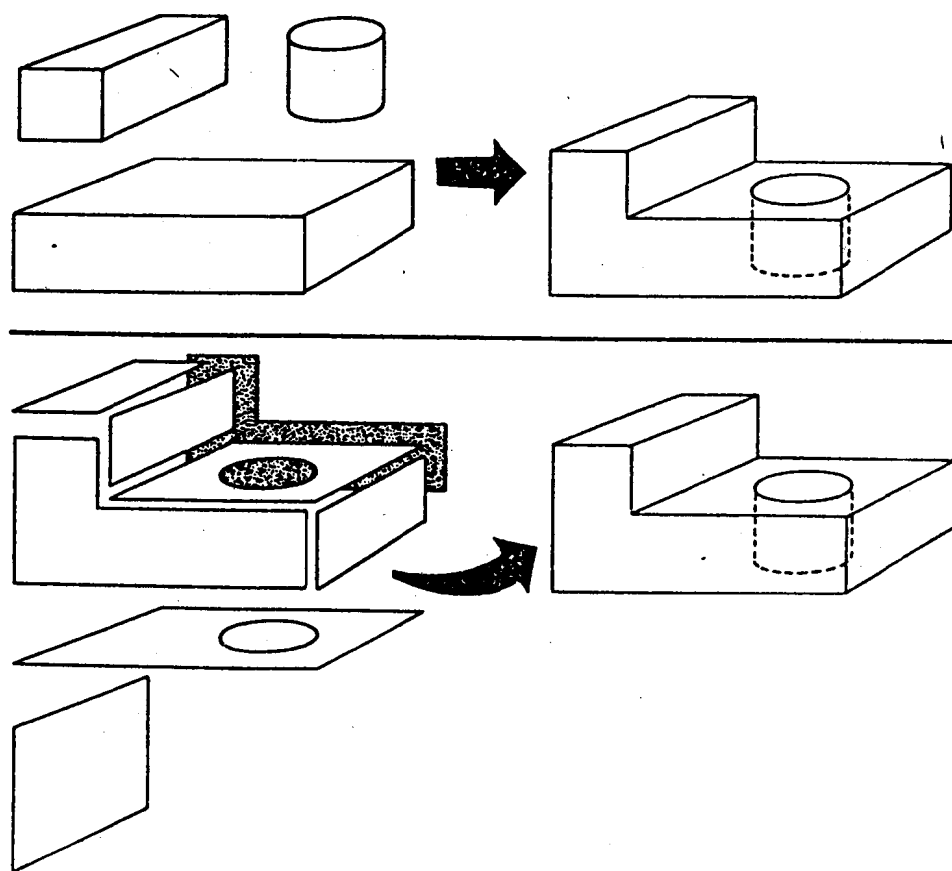


图 1—5

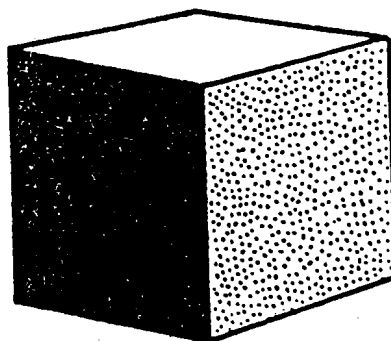
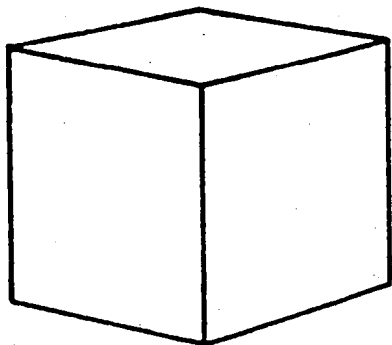
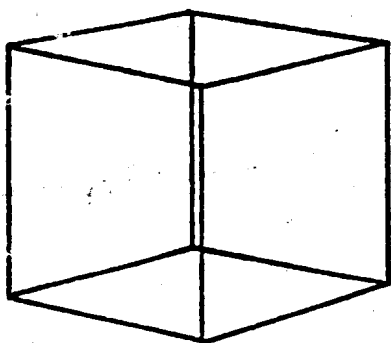


图 1—6

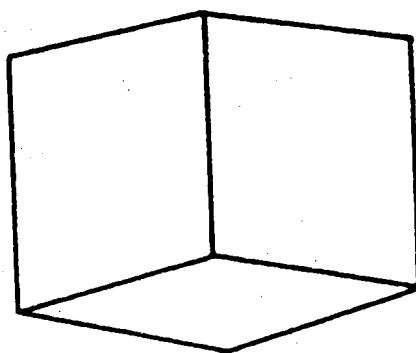
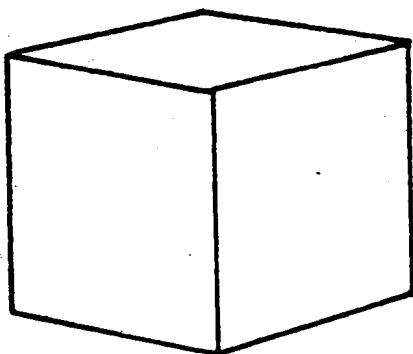
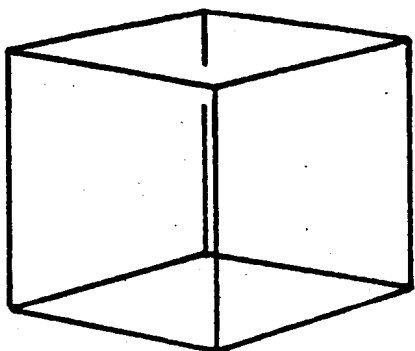


图 1—7

§ 1.8 显示选择

一个物体的一组世界坐标可以产生出不同的图象，如图 1—6 所示。最简单的图象是透视线框模型，这种模型是由线和顶点构成。该模型形状准确，但是没有消去隐面。在很多情形下，这种模式是有用的。因为你能看透模型，能确定画得是否正确，能确定内部部件的位置是否准确。边界描述方法是产生线框模型最有效的方法。

透视线框模型常遇到的一个问题是视图的不确定性，如图 1—7 所示。仔细观察上图，那么你的视觉感可能如中图所示，也可能如下图所示。

透视线框模型视图的不确定性可以通过去除看不见的面这个过程来消除，看不见的面叫做隐面，如何消除隐面，将在后面的章节里详细讨论。

第二章 三维模型处理

任何模型的世界坐标都描述了物体的形状。为了获得好的观察效果，在模型被透射到屏幕之前，必须对它作适当的处理。

§ 2.1 三维算法

图 2—1 显示的算式提供了模型处理的算法。如前一章所述，要将一个物体三维地显示在屏幕上，需经过三个步骤。首先，要确定物体的世界坐标 xyz ，建立物体的世界坐标数据库。其次，模型被旋转或平移到新的位置。旋转和平移的结果产生了一系列观察坐标 xyz 。旋转是基于球坐标，所图 2—2 所示。第三步：被旋转和平移了的三维模型被投影（通过投影公式）到显示屏幕上，产生 xy 平面坐标。

如果输入一组 xyz 世界坐标，通过图 2—1 的公式，将输出一组 sx 、 sy 显示坐标。

$$\left. \begin{aligned} xa &= cr1 \times x - sr1 \times y \\ za &= sr1 \times x + cr1 \times y \\ x &= cr2 \times xa + sr2 \times y \\ ya &= cr2 \times y - sr2 \times xa \\ y &= cr3 \times za - sr3 \times ya \\ y &= sr3 \times za + cr3 \times ya \end{aligned} \right\} \text{旋转公式}$$

$$x = x + mx; \quad y = y + my; \quad y = y + my \quad \text{平移公式}$$

$$\left. \begin{aligned} sx &= d \times x / z \\ sy &= d \times y / z \end{aligned} \right\} \text{投影公式}$$

图 2—1

§ 2.2 旋转公式

模型的旋转可以绕三个轴之一进行，即绕从上到下的竖直轴旋转，或绕从左到右的水平轴旋转，或绕从前到后的水平轴旋转。

首先讨论模型绕从上到下的竖直轴的旋转，如图 2—3 的上图示。在图 2—1 所示的公式中，变量 $sr1$ 表示旋转 $r1$ 角度的正弦， $cr1$ 表示旋转角度为 $r1$ 的余弦。 x 、 y 、 z 是世界坐标， xa 、 ya 、 za 是经这一转换的临时变量。

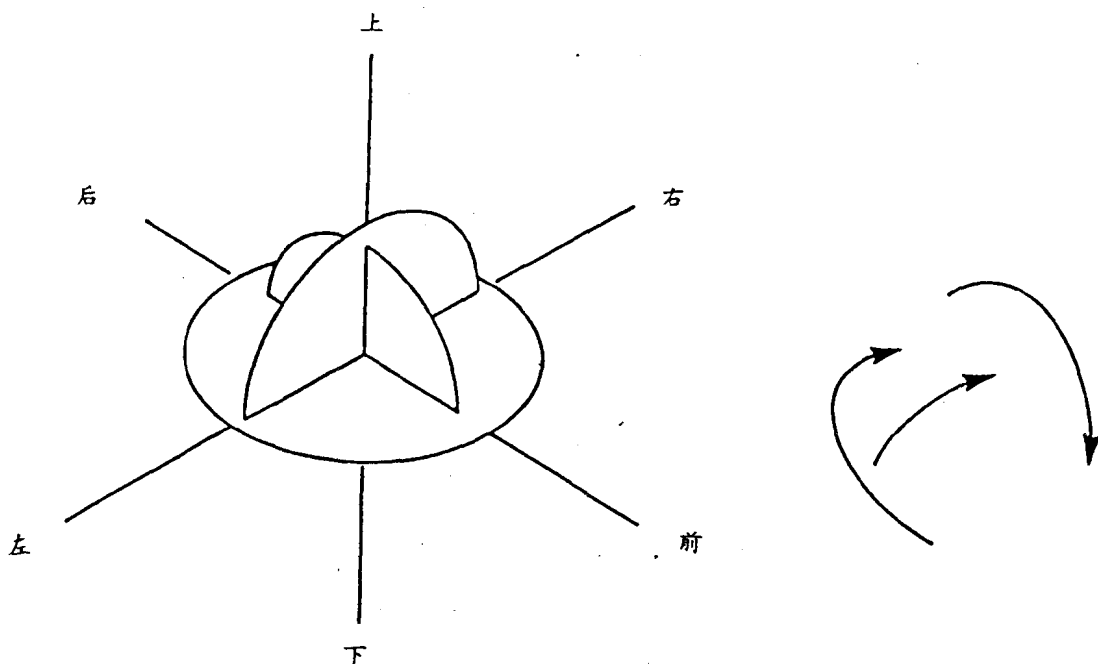


图 2—2

其次我们讨论模型绕从前到后的水平轴的旋转，如图 2—3 的中图示。旋转可以是顺时针方向，也可以是逆时针方向。图 2—1 所示的公式中，变量 $sy2$ 表示旋转角度 $r2$ 的正弦， $cr2$ 表示旋转角度为 $r2$ 的余弦。

最后我们讨论模型绕从左到右的水平轴的旋转，如图 2—3 的下图示。图 2—1 所示的公式中，变量 $sr3$ 为旋转 $r3$ 角度的正弦， $cr3$ 为旋转 $r3$ 角度的余弦。

通过调整这三种旋转角度的数值 $r1$ 、 $r2$ 、 $r3$ ，可以按照需要任意旋转模型，以期获得最好的观察效果。

§ 2.3 平移公式

对模型的进一步处理是模型平移。模型可以沿三维直角坐标 x 、 y 、 z 的任何一轴移动。因为观察者总是处在坐标原点 $(0, 0, 0)$ ，适当地平移模型，就使观察者相对模型处在合适的观察位置。

图 2—1 公式中的变量 mx 表示模型沿 x 轴（即从左到右）移动的距离， my 表示模型沿 y 轴（即从上到下）移动的距离。 mz 表示模型沿 z 轴（即从前到后或从近到远）移动的距离。图 2—4 显示这三种移动的效果。

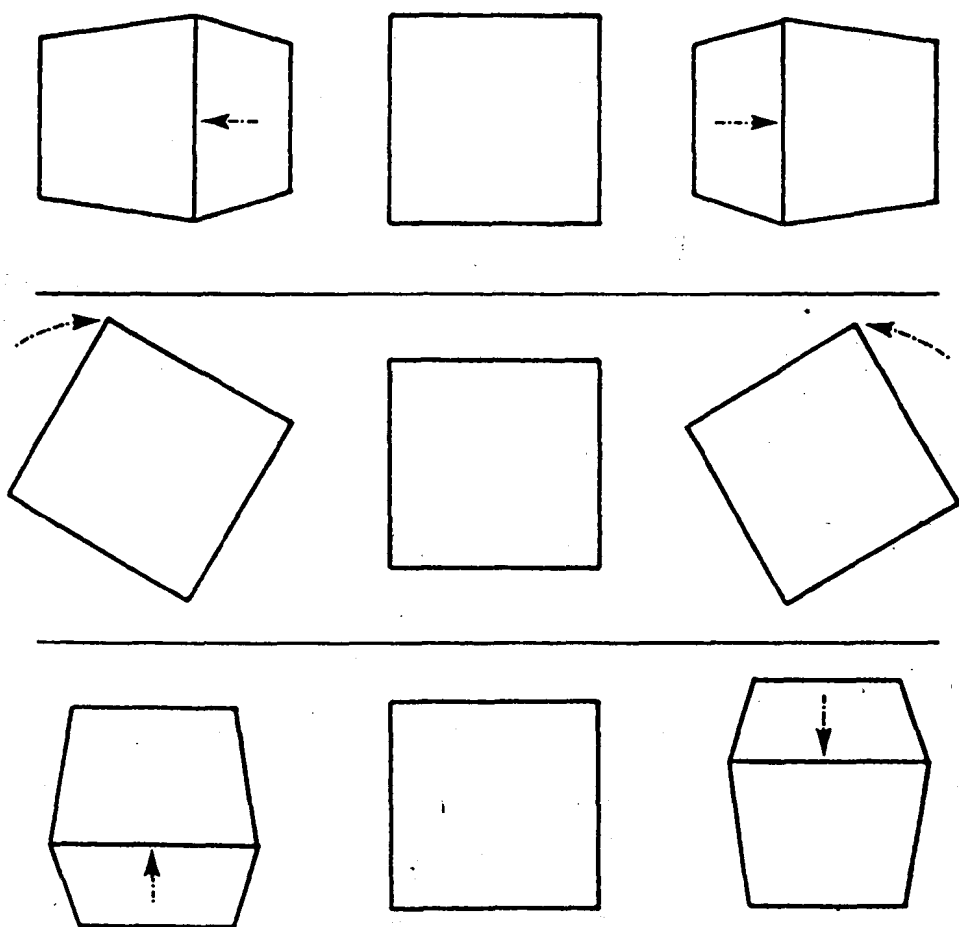


图 2—3

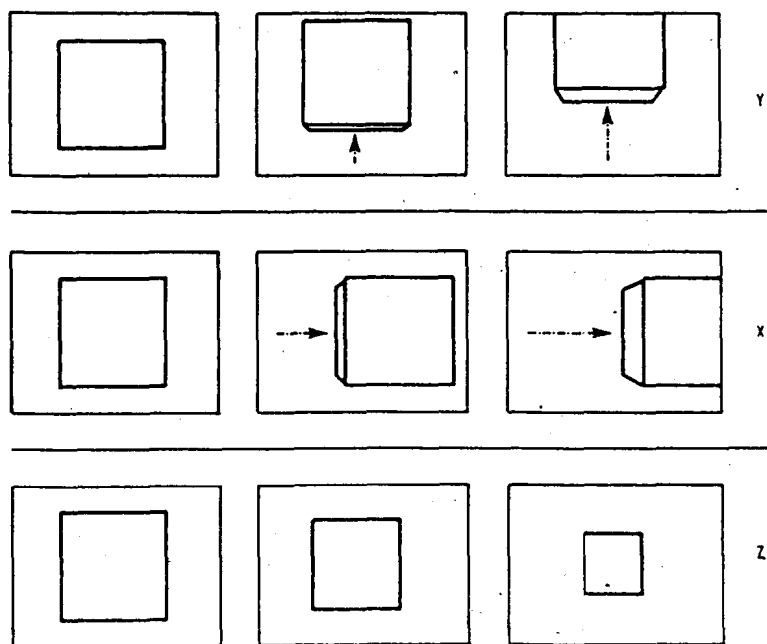


图 2—4